AMENDMENTS TO THE SPECIFICATION

Please delete the section entitled "SUMMARY" in its entirety and substitute the following section therefor:

SUMMARY OF THE INVENTION

The present invention provides a method and apparatus for avoiding a deadlock condition in a microprocessor having a BTAC that speculatively predicts target addresses of branch instructions that may cross cache line boundaries. In one aspect, the present invention provides a deadlock avoidance apparatus in a microprocessor having a speculative branch target address cache (BTAC). The apparatus includes a first signal, for indicating a miss of a fetch address in an instruction cache. The fetch address is a branch instruction target address speculatively provided by the BTAC. The apparatus also includes a second signal, for indicating an instruction formatter has determined the branch instruction wraps across two cache lines in response to decoding a first of the two cache lines. The instruction cache provides the first cache line containing only a first portion of the branch instruction prior to the first signal indicating the miss of the target address in the instruction cache. The apparatus also includes a third signal, for indicating the BTAC predicted the branch instruction is wholly contained within the first cache line, whereby a second of the two cache lines is not fetched because the BTAC predicted the branch instruction does not wrap across the two cache lines. The apparatus also includes a fourth signal for indicating execution logic has detected and corrected a BTAC misprediction. The execution logic does not detect the BTAC incorrectly predicts that the branch instruction is wholly contained within the first cache line because the instruction formatter is stalled waiting for the second cache line to be fetched. The apparatus also includes control logic, coupled to receive the first, second, third, and fourth signals, for invalidating the target address in the BTAC, in response to a true indication on the first, second, and third signals and a false indication on the fourth signal.

In another aspect, the present invention provides a pipelined microprocessor for avoiding a deadlock condition. The microprocessor includes an instruction cache, coupled to receive a fetch address. The microprocessor includes a branch target address cache

(BTAC), coupled to the instruction cache, for providing a speculative target address of a branch instruction in response to the instruction cache fetch address. The microprocessor also includes control logic, coupled to the BTAC, for invalidating the speculative target address in the BTAC in response to detecting a condition in which the speculative target address misses in the instruction cache after the instruction cache provides a first cache line in response to the fetch address, wherein the first cache line contains only a first portion of the branch instruction, and the BTAC incorrectly predicts that the branch instruction is wholly contained within the first cache line, thereby causing a second cache line containing a second portion of the branch instruction not to be fetched from the instruction cache. The microprocessor also includes an instruction formatter, coupled to the control logic, for decoding the first cache line and generating a signal to the control logic indicating the branch instruction is not wholly contained within the first cache line. The microprocessor also includes execution logic, coupled to the control logic, for detecting and correcting BTAC mispredictions. The execution logic does not detect the BTAC incorrectly predicts that the branch instruction is wholly contained within the first cache line because the instruction formatter stalls waiting for the second cache line to be fetched.

In another aspect, the present invention provides a method for avoiding a deadlock condition in a microprocessor having an instruction cache and a speculative branch target address cache (BTAC). The method includes generating a speculative target address of a branch instruction partially contained in a first cache line provided by the instruction cache in response to a first fetch address, in response to applying the first fetch address to the BTAC. The method also includes providing the target address as a second fetch address to the instruction cache without fetching a next cache line sequential to the first cache line, in response to the BTAC predicting the branch instruction is wholly contained in the first cache line. The method also includes determining the BTAC incorrectly predicted the branch instruction is wholly contained in the first cache line in response to decoding the first cache line. The method also includes detecting a miss of the target address in the instruction cache. The method also includes stalling an instruction formatter waiting for the next cache line sequential to the first cache line to be fetched, thereby preventing execution logic configured to detect and correct BTAC mispredictions from detecting that the BTAC incorrectly predicted that the branch instruction is wholly contained within the first cache line. The method also includes invalidating the target address in the BTAC, in response to the determining and the detecting.

In another aspect, the present invention provides a computer program embodied on a computer-readable medium, comprising computer-readable program code for providing a microprocessor. The program code includes first program code for providing an instruction cache, coupled to receive a fetch address. The program code includes second program code for providing a branch target address cache (BTAC), coupled to the instruction cache, for providing a speculative target address of a branch instruction in response to an instruction cache fetch address. The program code also includes third program code for providing control logic, coupled to the BTAC, for invalidating the speculative target address in the BTAC in response to detecting a condition in which the speculative target address misses in the instruction cache after the instruction cache provides a first cache line in response to the fetch address, wherein the first cache line contains only a first portion of the branch instruction, and the BTAC incorrectly predicts that the branch instruction is wholly contained within the first cache line, thereby causing a second cache line containing a second portion of the branch instruction not to be fetched from the instruction cache. The program code also includes fourth program code for providing an instruction formatter, coupled to the control logic, for decoding the first cache line and generating a signal to the control logic indicating the branch instruction is not wholly contained within the first cache line. The program code also includes fifth program code for providing execution logic, coupled to the control logic, for detecting and correcting BTAC mispredictions. The execution logic does not detect the BTAC incorrectly predicts that the branch instruction is wholly contained within the first cache line because the instruction formatter stalls waiting for the second cache line to be fetched.

In another aspect, the present invention provides a computer program embodied on a computer-readable medium, comprising computer-readable program code for providing a deadlock avoidance apparatus in a microprocessor having a speculative branch target address cache (BTAC). The program code includes first program code for providing a first signal, for indicating a miss of a fetch address in an instruction cache, wherein the fetch address is a branch instruction target address speculatively provided by the BTAC. The program code also includes second program code for providing a second signal, for indicating an instruction formatter has determined the branch instruction wraps across two cache lines in response to decoding a first of the two cache lines, wherein the instruction cache provided the first cache line containing only a first portion of the branch instruction prior to the first signal indicating the miss of the target address in the instruction cache. The program code also includes third program code for providing a third signal, for indicating the BTAC predicted the branch instruction is wholly contained within the first cache line, whereby a second of the two cache lines is not fetched because the BTAC predicted the branch instruction does not wrap across the two cache lines. The program also include fourth program code for providing a fourth signal, for indicating execution logic has detected and corrected a BTAC misprediction, wherein the execution logic does not detect the BTAC incorrectly predicts that the branch instruction is wholly contained within the first cache line because the instruction formatter is stalled waiting for the second cache line to be fetched. The program code also includes fifth program code for providing control logic, coupled to receive the first, second, third, and fourth signals, for invalidating the target address in the BTAC, in response to a true indication on the first, second, and third signals and a false indication on the fourth signal.

An advantage of the present invention is that it enables proper program operation in a microprocessor that employs a speculative BTAC and that executes instructions that can cross cache line boundaries.

Other features and advantages of the present invention will become apparent upon study of the remaining portions of the specification and drawings.